



OpenBIM-Tango integrated virtual showroom for offsite manufactured production of self-build housing

Farzad Pour Rahimian^{a,*}, Veselina Chavdarova^b, Stephen Oliver^b, Farhad Chamo^b

^a Department of Mechanical and Construction Engineering, Faculty of Engineering and Environment, Northumbria University, Newcastle NE1 8ST, UK

^b Innovative Construction Technologies and BIM Research Cluster, Department of Architecture, University of Strathclyde, Glasgow G1 1XW, UK

ARTICLE INFO

Keywords:

openBIM
IFC
Interoperability
Constraint-driven design
Immersive technologies
Agnosticism
Synchronised concurrent user accessibility
Low-latency

ABSTRACT

As a result of progressive use of BIM in the AEC sector, the amount of diverse project information is increasing rapidly, thus necessitating interoperability of tools, compatibility of data, effective collaboration and sophisticated data management. Media-rich VR and AR environments have been proven to help users better understand design solutions, however, they have not been quite advanced in supporting interoperability and collaboration. Relying on capabilities of openBIM and IFC schema, this study posits that this shortcoming of VR and AR environment could be addressed by use of BIM server concept allowing for concurrent multiuser and low-latency communication between applications. Successful implementation of this concept can ultimately mitigate the need for advanced technical skills for participation in design processes and facilitate the generation of more useful design solutions by early involvement of stakeholders and end-users in decision making. This paper exemplifies a method for integration of BIM data into immersive VR and AR environments, in order to streamline the design process and provide a pared-down agnostic openBIM system with low latency and synchronised concurrent user accessibility that gives the “right information to the right people at the right time”. These concepts have been further demonstrated through development of a prototype for openBIM-Tango integrated virtual showroom for offsite manufactured production of self-build housing. The prototype directly includes BIM models and data from IFC format and interactively presents them to users on both VR immersive and AR environments, including Google Tango enabled devices. This paper contributes by offering innovative and practical solutions for integration of openBIM and VR/AR interfaces, which can address interoperability issues of the AEC industry.

1. Introduction

The implementation of Building Information Modelling (BIM) has quickly gathered momentum for its promise of structure, organisation and efficiency of the work processes and tools. The benefits of BIM for helping integration of architectural, construction or manufacturing projects have been widely highlighted in the seminal literature [1,3,8,14,38]. Notwithstanding these paybacks, there is an inherent flaw in the structure of this semantically rich modelling technology. The very accumulation of diverse data from different disciplines through consecutive phases increases the file sizes and structure complexity, thus impeding the smooth flow of information exchange and archiving interoperability [3] in these models. Sophistication of recent BIM models exceeds the comprehension of the stakeholder, clients, end-users, owners and other non-engineering professionals preventing them from being effectively involved in various design development and project implementation review processes [40], due to the so called

“black-box effect”, which refers to a system without transparency and enough legibility for everyone.

Advanced visualisation could potentially address this problem, by further enhancing comprehension of non-technical people from naturally technical BIM models. Yet, despite improvements in orthogonal drawings, current BIM technologies are primarily designed as management tools or repositories of interrelated descriptive and 3D information, so there are limitations in their visualisation capacities. Virtual and Augmented reality (VR and AR), however, are widely proven as enabling technologies to address these issues, by helping clients experience design in 3D space, assess indoor factors such as lighting, plan for future maintenance and decide for themselves whether it would meet their needs [9,15,21,34,35]. Conversely, currently available immersive experience solutions on the market offer only partial opportunities for building design integration; or otherwise require advanced technological skills from the user. As a result, users with no prior advanced training and the necessary hardware are limited in

* Corresponding author.

E-mail address: farzad.rahimian@northumbria.ac.uk (F. Pour Rahimian).

<https://doi.org/10.1016/j.autcon.2019.02.009>

Received 19 March 2018; Received in revised form 4 February 2019; Accepted 12 February 2019

Available online 21 February 2019

0926-5805/ Crown Copyright © 2019 Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

what they can view and achieve with these packages. Additionally, the existing software solutions offer no integration of BIM, hence, no data on the construction materials, services or costs is available for interaction.

As a response to this functionality gap between BIM and immersive technologies, this study proposes an interface that integrates the two, so as to streamline design processes and provide a comprehensive pared-down BIM system. The aspiration of this interface is to be fully agnostic towards the diverse BIM editing tools, which can become a source of input and offer synchronised concurrent user accessibility with low latency that can promote active collaboration. To illustrate the operational principles of this system, the project is based around offsite manufactured self-build housing, where the manufacturing company would present house designs in a virtual environment directly from their BIM models, allowing their clients to walk through and customise a range of home features remotely.

2. Related studies

This study was concerned with four groups of literature to form theoretical and methodological basis for its developments. These four areas are as follows: 1) OpenBIM Standards in order to provide the project with reliable open source platform to facilitate unlimited coding for integration of BIM models with other environments, i.e. VR and AR; 2) Parametric and generative design interfaces as opposed to the traditional geometric CAD tools and their impact on how designers think and collaborate with each other; 3) The emerging VR and AR technologies and their role in facilitating intuitive and immersive interfaces for users to interact with the design solutions without the need for higher level of skill and discipline knowledge for understanding design solutions, collaborate with the professional team members and contribute to decision making throughout the project life cycle; and 4) The impacts of these high-tech technologies on people involved in the teams and the ways through which these technologies change people's interactions with each other. The following subsections elaborate on these four areas and present how seminal literature approached these issues.

2.1. OpenBIM standards and the IFC format

The exponentially rising interest in BIM technologies in the Architecture-Engineering-Construction (AEC) industry has led to an acute necessity for a common vehicle of exchange between disciplines and effective management of vast quantities of data. Research has already noted a lack of interoperability as one of the main obstacles in the adoption of BIM in the AEC industry [40], where the role of BIM as an enabler of efficient collaboration and communication is hindered by the diversity and heterogeneity of the output information. The implementation of a universal platform for referencing processes and information throughout the life cycle of an asset could potentially eradicate the issues related to software incompatibility. As an attempt to tackle this issue, the buildingSMART alliance developed the openBIM approach based around open standards and formats [33]. Whilst the emphasis in buildingSMART International is on data and technology, less so on people and processes [3], the openBIM concept initiates a collaborative approach among a wide range of disciplines, where different users can use specialised software and store diverse data in the same unified model. The aim of this movement was to establish a common platform for sharing information in a common language, enabling participation and engagement in the project regardless of the tools employed by different practitioners [39].

The Industry Foundation Classes (IFC) format is the primary vendor-neutral data model developed by buildingSMART and the majority of the BIM authoring software support the IFC import/export feature. The IFC schema is an extensible object-oriented data model divided into base entities and sub-entities. This format classifies and gives structure to the data allowing different sets of information to be extracted easily.

The inherent data consistency of the format aims to eliminate the need to input the same information multiple times, thus minimising further error and allowing the project team to focus on the compatibility of their workflows.

This common IFC structure has potential to serve as a vehicle to achieving agnosticism towards BIM editing tools where team members and companies do not need to be selected based on the software they use but on their actual knowledge and capabilities. The different needs of the professionals involved in a project throughout its life cycle necessitate multiple types of geometry, properties and relations so the IFC construct preserves the data structures inherited from the native format - geometry, logical relations, and attributes of the 3D elements. However, in order to accommodate for all of these, the IFC format is highly redundant. Consequently, coordinating and communicating building data becomes increasingly complicated, despite the compatibility of formats. The resulting high complexity leads to lower usage among professionals where the limited understanding of import and export features causes translational errors and impedes the intent for interoperability [2,33]. As a response to this inherent dysfunction of the IFC exchange, many studies [e.g.,10,11,22] proposed various code checking logics to ensure open BIM interoperability and data consistency in order to minimise BIM data loss when transferring to IFC data structure from one library to another.

2.2. Parametric and generative design

The arising complexity of the projects in the AEC industry increasingly requires advanced skills to interact with design tools resulting in low engagement of architects and engineers with end-users, clients, owners and other professionals with partial spatial training. The consequent reduced input from key contributors results in design flaws and construction errors that then accumulate cost implications in the execution of the project. BIM was introduced to improve the efficiency of the AEC industry by providing means for accurate measurements and access to a comprehensive body of information, so it laid the groundwork for better organisation, development and procurement of building projects. The main advantage of BIM is that the information directly relates to 3D elements, making it easier to comprehend and visualise. As a result, the object-based, data-rich environment of BIM opens the possibility for quick and interactive methods of design generation and development [7,32,44].

Oxman [31] discussed a new paradigm in design thinking, where the technological advances and their dissemination within the industry bring together cognitive and computational efforts in the design process. The algorithmic reasoning, based around rules and instructions, is a way of thinking that creates form. Writing code as a means of building design expression, however, requires not only spatial comprehension but also an understanding of the formal expression of design actions and visual outcome in code. The design process is no longer made through single decisions about individual objects or processes, but rather through a matrix that simultaneously covers a larger aspect through a network of systematic relationships.

Parametric modelling techniques, based on constraints and variables, define and diversify the options for design through different parameters (e.g. materials and energy performance). Predefining the logical relations between geometric entities allows them to move or change as per the values of the established parameters or generate random iterations. This then ensures further flexibility in the design process and enables the practitioner to choose the most suitable of the generated options.

In addition to optimising the performance of building elements, parametric design holds great potential in improving the efficiency of materials use and reducing waste [12,28], addressing issues related to resource constraints [25], supporting prefabrication as a means of optimising production, transportation and installation [20,26], and modular coordination [5]. Yuan, et al. [44] noted though that

prefabricated building design is not fully integrated in BIM products and libraries, thus hindering the potential of parametric design. In response, their study focused on establishing an information model suited for manufacture, so they introduced the concept of parametric design oriented to DfMA (Design for Manufacture and Assembly) and indicated the capacity of the IFC format as an exchange vehicle.

2.3. Mobile interfaces and immersive technologies

In addition to complex technical issues, the AEC industry is also facing issues related to the engagement of customers and stakeholders who might have no prior technical training, and consequently are unable to grasp and visualise the solutions to spatial problems [26]. Despite BIM's great potential for data accumulation and organisation, it only offers limited immersive capabilities within building designs. Therefore, an additive technology, such as AR or VR, is required to maximise the potential of data loss within BIM for better visual comprehension. The increasing tendency towards integration of these technologies is also aimed at increasing labour productivity and efficient use of information [24]. Real-time interaction through AR or VR allows the user to transcend from their physical location into the environment of the entire project, which then opens a possibility for more effective interaction and simulation analysis of projects [41].

Effective client engagement is particularly important in housing projects as the non-professional end-users often have limited spatial comprehension and are unable to fully understand conventional architectural drawings, which is crucial for their satisfaction of the final product. Damen, et al. [13] recognised the need for participatory design to support decision-making and the potential of BIM's parametric capabilities as a vehicle for integrated design development of collective self-organised housing. Their approach to client inclusion is based on a seamless translation of parametric design in physical world through AR/VR interfaces. Wang, et al. [42] also investigated the possibility for a conjunction between BIM and AR, so as to provide an on-site tool that integrates BIM data in a physical context thus helping users construct mental models of the building design. Additionally, Ren, et al. [36] facilitated architects and engineers during construction and developed a mobile AR system that integrated lightweight BIM data and 3D registration.

Whilst AR is an effective way of overlaying digital information over real physical context and identifying clashes with existing structures on site, VR also holds great promise for end-users' engagement as it is a powerful tool for visual communication of building designs and a dynamic feedback initiator. Rasmussen, et al. [35] developed a case study for furnishing design that uses BIM as a visualisation tool and integrates it into VR to assist communication and decision-making exemplified in a design workshop as an initiating step towards collaboration between designers and users. Despite the various attempts to bring together BIM and immersive technologies, compatibility of formats and workflows remains an obstacle to progress. Du, et al. [16] recognised the issue of interoperability between BIM data and VR tools, as well as the resulting data transfer difficulty. Their research focused on the necessity to synchronise information and ensure optimal decision-making, so they developed a cloud-based synchronisation system that maps BIM meta-data from Autodesk Revit, sends it to a server where it is checked against and exchanged with information from a VR game engine.

2.4. Media richness and social presence

Effective communication management is crucial for the success of construction projects, a critical skill for design professionals [29] and cornerstone function of project management [37,43,45]. Conversely, ineffective communication has been identified as a source of conflict in construction projects [19,27], including communicative ineffectiveness in the form of poor documentation and lack of trackability [23]. In their respective, cross-disciplinary qualitative assessments of the causes behind construction project delay Odeh and Battaineh [30] and Assaf and

Al-Hejji [4] found that although perceptions of primary sources of delays vary by practitioners' roles in the construction industry, communication failures play a critical role in overrun.

BIM was a major step forward in how communication is practiced in the industry, improving efficiency, openness, accountability and the richness of the information transferred between parties. However, it has not made significant changes to the immediacy and latency aspects of communication. Its richness cannot be fully utilised at level two of BIM. BIM level three will facilitate greater application of current Computer Mediated Communications (CMC) technologies, however for innovation to flourish it should be based on an open standard rather than merely accommodating post-process generation of open standard files. The project presented in this paper does not currently solve all these problems, it does however serve as a foundation for developers to extend their preferred interface whether Revit or MS Word and integrate other components.

3. Project framework

The first phase of the project development involved an extensive literature review, to diagnose the current issues in the AEC industry and identify the enabling characteristics and limitations of the technologies that have already been employed in the industry. This study served to determine potential routes on how different technologies can be utilised in conjunction to achieve smoother collaboration and higher levels of interoperability. Interoperability and portability ensure innovation and optimal market growth in software and platform development. It is no longer a nicety to be included at the discretion of the vendor [18]. These characteristics prevent vendor-lock from stifling innovation and bottlenecking industries into implicit arrangements with prominent market leading vendors better ensuring smaller and indirect service providers are better incentivised by reduced risk to innovate. Blind [6] noted that standardisation can promote commercial innovation and provide a platform for researchers and other actors to diffuse and expand on ideas. Vendor-lock on the other hand can always be counter-productive for providers and displeasing to users [18]. Gasser and Palfrey [17] argued that the internet and emails are prime examples of why interoperability and standardised development interfaces are beneficial to all and leverage innovation, by providing tools for anyone to develop tools without vendor constraints.

There is little previous work on concurrent multiuser, low-latency BIM synchronisation, whereas theoretical studies investigated the desirability, necessity and challenges with regards to developing such systems. Du, et al. [16] developed a Revit API-based solution, similarly linking vector CAD with the Unity discrete environment. Liu, et al. [26] also used Revit API for leveraging rich information in BIM model into an algorithm in order to optimise panel use and panel cutting in offsite construction. Their project differs from the project discussed in this paper in both implementation and targeted features, though these project are complementary.

Whilst the framework developed in this study is closer to desirable interoperability features through IFC-Centric, both studies mentioned above proposed Revit integration with intrinsic model integrity and unidirectional constraint validation, which are yet to be achieved in the project presented in this paper. Generally, their solutions provide better integration with existing systems at the expense of interoperability and bidirectional interactions, and better asset distribution at the expense of transaction loads. Contrarily, this project chose focusing on openBIM principles and IFC, rather than a vendor specific product. Nonetheless, this paper acknowledges their attempts as a significant advancement in BIM integration research. The resulting project highlighted important considerations, alluding this research in developing new solutions. As such, the aim of the present study was developing a method for integration of openBIM data into immersive environments.

Upon this scientific basis, the second phase of the project focused on developing an integration strategy for the openBIM, VR and AR

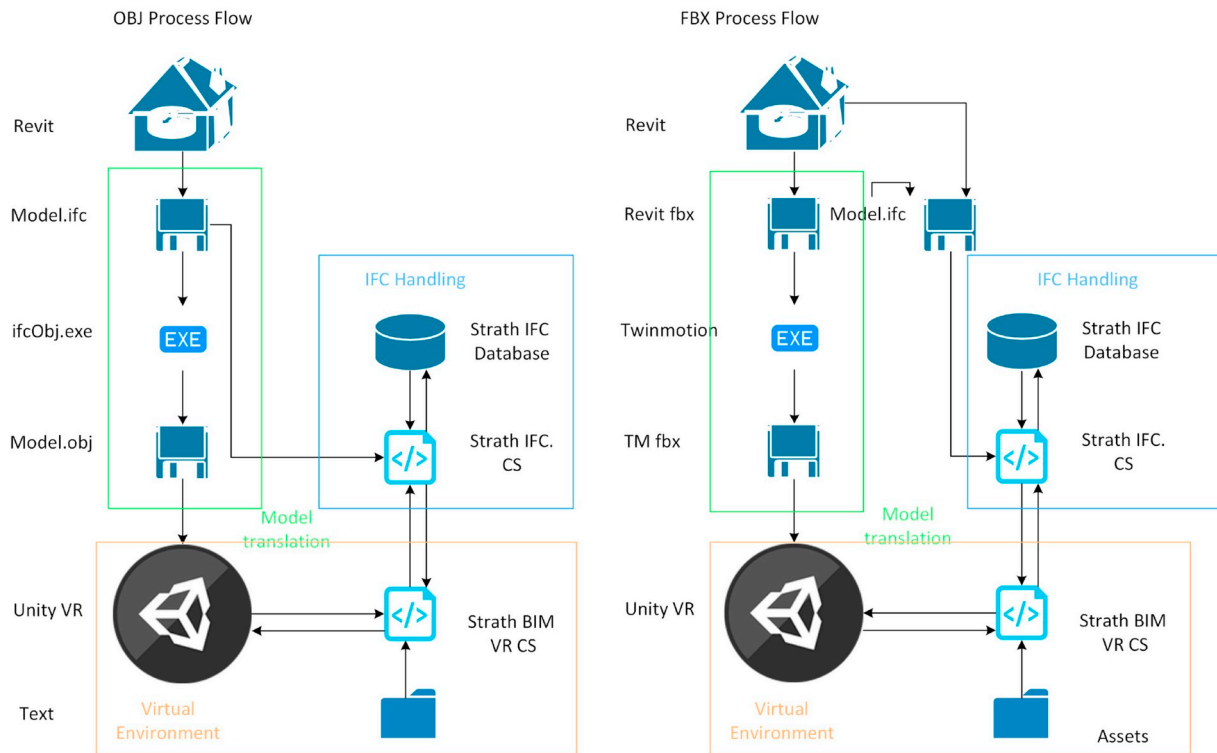


Fig. 1. Process flows for BIM data conversion and integration within virtual environment.

technologies. The most suitable formats for the purposes of the project were determined through preliminary technical tests. The third phase was devoted to the prototype design, where the framework development adopted an agnostic approach towards openBIM editing tools using the vendor neutral IFC file format. An iterative process of testing, reviewing and troubleshooting ensured design fixity and validated the framework of interoperability. Finally, feedback data collection was conducted via usability testing. The potential impact of the integrated virtual showroom prototype was identified through revisiting the literature review and evaluating the results of the project in the context of offsite manufactured production of self-build housing.

4. Prototype development

4.1. Project overview, BIM library and IFC database

The virtual showroom prototype was developed in consecutive stages beginning with the 3D modelling of a kit home in BIM environment. The project used Norscot Joinery's kit home design called "Assynt 4-Bed", which was the main industry partner and one of the funders of this project. In the following sections of this paper, Norscot Joinery Ltd. will be referred to as Norscot. Full design information on the Norscot Assynt 4-Bed was obtained from floor plans, elevations and rendered images. Given Norscot's wide portfolio of products, the BIM library of the project was expanded to include Norscot doors, windows and wall types, modelled in 3D as Revit families.

Subsequently, the modelled BIM assets were integrated into a game engine environment, where they were coupled with interactive functionalities that allow the user to explore and manipulate building elements within 3D space. The game engine setting containing all features and mechanics was then split into two different flows – a virtual reality application and an augmented reality interface which benefitted from recently developed Google Tango technology. That allowed for two different immersive experiences of the home design - 1) in virtual space through a desktop device and a VR headset, and 2) in augmented reality through a Tango enabled mobile device.

The BIM model and its library served as the basis of an IFC database created using PostgreSQL. The IFC objects were linked to Unity as meshes through an ifc.cs module, where a method for converting triangle meshes into faces was developed including an experimental mode for identifying joined faces. This method will prove invaluable if energy simulation integration occurs in later research.

BIM integration facilitates building model modification indirectly through Unity's environment using the purpose-built C# BIM library for IFC models. The feature was designed to bind BIM and Unity building models such that information from the BIM model could be retrieved for Unity environment equivalent objects and interactions with furnishing or materials in Unity would be reflected in the BIM model. Development was carried out in C# for both Unity and IFC library components. The IFC file format was chosen for interoperability between any environment referencing the library and all BIM packages implementing ISO 16739:2016, in contrast to existing software collectively sharing similar functionality with limited interoperability through development with vendor-specific APIs. Interactions between Unity and IFC geometry were delegated to an intermediary environment for translation between Unity's discrete and IFC's continuous/functional entity definitions which provides data structure and coordinates system resolution of differing environments.

4.2. System architecture

Previous research looking at how to use BIM and VR technologies in conjunction has been only partial or inconclusive in its strategic approaches leaving gaps in the conceptual frameworks. Once the appropriate software and equipment were identified, the next critical choice was to select the most suitable format that would allow a transition from BIM to VR with a minimum loss of data and geometry. The flexibility of the chosen file formats that transfer 3D information, as well as the appropriate handling of the data are key in the successful integration of BIM models into game engine environment. Autodesk Revit offers a reasonable range of output formats, yet certain technical challenges marked the transition of the 3D model to VR environment

for further work on the virtual showroom. Fig. 1 illustrates the process flows for BIM data conversion and integration within virtual environment that were adopted in this project.

The initial strategy for BIM integration into VR was based on Autodesk's FBX format. The 3D model was exported before being delivered to Unity for VR functionality development. Materials mapping was the main issue in that conversion process, since the FBX import appeared not to retain any textures when loaded into the game engine environment, so brick, wood, glass or any other material would visualise as the same plain white solid surface. This meant that the underlying causes and potential solutions had to be further investigated in detail. Given that BIM and VR technologies are not necessarily developed for the same purposes, they treat files and materials differently. Revit, a modelling and editing tool, uses Autodesk or user-created material libraries stored in a specific directory on the user's computer, whilst Unity references materials from the project. Consequently, when an FBX file is imported into Unity, it does not automatically carry over the materials from the application that created it.

Another issue was Unity's primarily use of interdependent relative coordinate systems which requires foreknowledge of the imported models since relative systems don't always correlate with other nests. Unity's XYZ coordinate system causes several issues requiring mesh coordinates and world coordinates to be handled with separate tools. Additionally, imported objects may be bound to pivot points, thus making world translations ineffective. As a method of floating-point error mitigation, scaling differs between vector and unity environments. IFC conversion to an OBJ format is susceptible to mild corruption, when translating faces which will need to be considered a limitation when attempting to work with geometry. BIM formats are not created to be bidirectional, hence, precision of coordinates must be retained, or Revit will not be able to reconstruct the building. Database transaction latency and design complexity led to the development of the inline C# model for IFC interactions (Fig. 2).

4.3. BIM server

This study created a proof-of-concept IFC BIM server (Fig. 3) for concurrent multiuser, low-latency communication between interfacing applications utilising the companion IFC BIM client which resolves numerical data structure uncertainty and IFC modification concurrency through a DSL. The server manages multiple IFCFile instances with command history tracking for incremental partial or complete updating, without replacing the source IFC file. The impetus for development beyond agnostic BIM interfaces creates a framework for high media richness in virtual environments that can benefit professional use.

The server provides a framework for connecting multiuser concurrency and data concurrency between interfacing applications, implementing the BIMClient, accommodating multi-IFC instance management. It is designed to accommodate development of BIM interfaces in most applications with APIs. Connected applications are allocated

read/write named pipes (BIMServerConnection). These are assigned to the IFCFile instance's associated IFCCConnection through an IFCCCommand.Load request.

Concurrency between server and client instances is managed through transaction histories and the pipe locking system, such that interactions from an interface are approved and applied centrally prior to distribution between other clients. Messaging between clients and interfaces does not implement any form of serialisation, instead using IFCCCommand transactions, which replicate actions applied to local IFCFile instances with the central server instance serving as mechanism for maintaining consistency between the server and clients.

In summary, IFCFile instances are created on both server and client such that interactions with the BIM model are not bottlenecked from piping or locking, transaction histories are tracked such that an out of date model can, within reason, be updated to the current state of the building model without importing and exporting through an ISO 10303 compliant application. Interaction integrity is ensured through IFCCCommand messaging, rather than serialisation. This permits decoupled incremented updates to local models and data type integrity for 128bit real numbers regardless of interface data structure.

Linking clients to models and distributing command from the server is achieved through unique keys assigned to the IFCCConnection and BIMClientConnection. When the server receives a message from a client, it uses the Message.Parameters.ClientKey to identify the linked IFCCConnection. Once it is identified, the server distributes a message with Message.Parameters.Lock active message to each linked client, before validating and applying the IFCCCommand. Once clients receive a lock request, the connection write pipe is disabled. In contrast to the client, server message interpretation and application—whilst threaded for each IFCCConnection—is synchronously processed by disabling message retrieval during processing. However, server message distribution post-process is not processed in the same way, therefore is dependent on sequential message transmission. Fig. 4 describes the BIM server message processing flow.

Commands from interfaces are interpreted by the BIMServer instance and applied through IFC namespace models. A simple DSL was created for messages in place of serialisation to reduce network loads, standardise interactions between interfaces and IFCFile models. This method maintains consistency with IFC real numbers which are sensitive to manipulation beyond precision loss and will break a model if adherence to their rules is not strictly maintained.

4.4. IFC model

IFC data is stored in an EXPRESS schema model represented in single record entries with non-sequential unique IDs. Record associations are predominantly tracked as reference ID lists in the parameters representing a loosely coupled graph structure of coordinates, entities, functions, layers and properties however, some property associations are linked through string properties or parent associations. The record defines three generic values, record ID, category and parameters where the category identifies the type of object and dictates parameter types. Data is ASCII encoded resulting in a very large file compared to binary CAD representations.

There are two primary physical entity definition structures: coordinate-based planes and profiles. The former type may share points and materials with similar objects, whereas the latter does not share points but may share materials. Materials are stored by name only and not exported with the model. Any importing interface must have an associated material to render the model, as it is in the interface that created it. Colours related to materials are stored as separate records which may be shared. Entity-specific coordinate systems may share localisation. World coordinates and axis direction vectors whose axes are not necessarily parallel to those of the world coordinate system or other entities, profile entities have secondary direction vectors to orientate extrusions. The model structure supports revision histories. Fig. 5 shows an IFCBEAM graph.

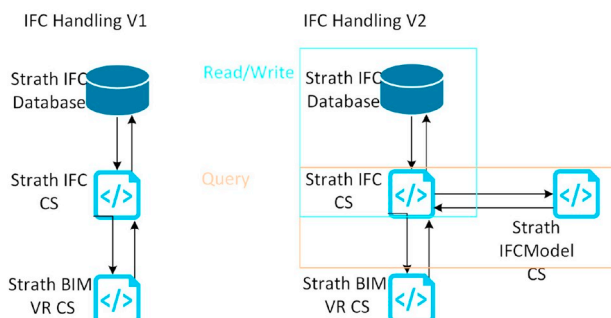


Fig. 2. File handling flow options.

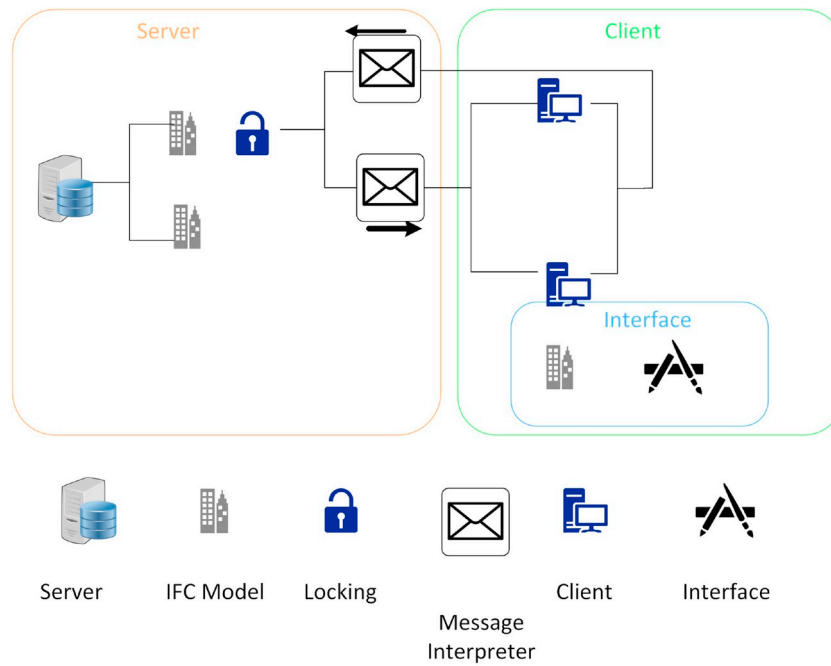


Fig. 3. BIMServer/BIMClient relationship model.

4.5. Unity planar geometry

Unity, in addition to using a discrete world, is not designed for planar geometry and subsequently not ideal for several functionalities of the prototype. Game engines including Unity, almost ubiquitously use some form of discrete polygon mesh, typically triangle meshes, for both rendering and associated data structures. Beyond hardware specialisation in working with triangles, they have convenient properties which accommodate efficient rendering and raycasting. Triangle vertices are coplanar reducing operation complexity and accommodate using barycentric coordinate systems. Among other things and in terms of computer graphic engines, these systems are most notably useful for interpolation and efficient point-in-boundary calculation.

Triangles, however, are not well suited to the operations required for this project. Their ignorance of surface relationships, as well as the decoupling between vertices and triangle definitions prevent linking to IFCFACEs. They hinder profile-dependent evaluation including surface

area calculation and define voids as an absence of triangles rather than closed boundaries. These features are fundamental to partition rendering modifications since Unity materials are shared with the mesh rather than surfaces, interface attachment and parametric modelling. These in turn require knowledge of surface relationships.

This research resolved this through the creation of the IFC library's Facer, which constructs faces from vertex and triangle arrays, identifies boundary polyline, shared vertices and surface relationships retaining the game engine associations between both data structures. By creating the relationships between Unity and IFC.Facer.FaceSets this study was able to develop the desired functionality whilst taking advantage of triangular mesh properties. This was achieved through a simple edge detection algorithm translating triangles into Face objects. The process involved identifying coplanar and connected subsets using shared vertices to find linked line segments. Then, segments present on two triangles were filtered out and segment vertex associations were used to create closed polylines. Surface voids were identified by repeating the

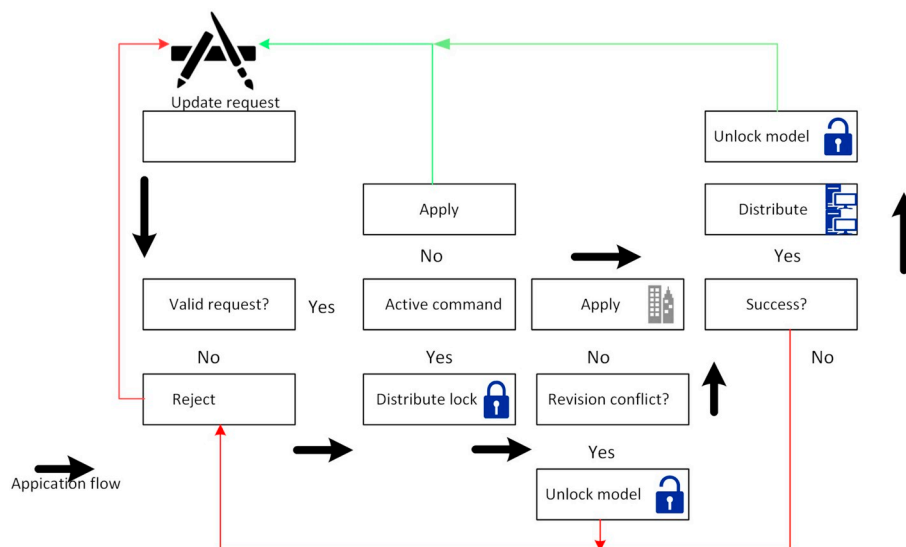


Fig. 4. BIM server command transaction model.

```

383336 IFCBEAM
41 IFCOWNERHISTORY
38 IFCPERSONANDORGANIZATION
35 IFCPERSON
37 IFCORGANIZATION
5 IFCAPPLICATION
1 IFCORGANIZATION
383306 IFCLOCALPLACEMENT
122 IFCLOCALPLACEMENT
32 IFCLOCALPLACEMENT
383366 IFCLOCALPLACEMENT
383365 IFCAXIS2PLACEMENT3D
6 IFCARTESIANPOINT
31 IFCAXIS2PLACEMENT3D
6 IFCARTESIANPOINT
121 IFCAXIS2PLACEMENT3D
6 IFCARTESIANPOINT
383305 IFCAXIS2PLACEMENT3D
383303 IFCARTESIANPOINT
19 IFCDIRECTION
13 IFCDIRECTION
383334 IFCPRODUCTDEFINITIONSHAPE
383331 IFCSHAPE REPRESENTATION
100
IFCGEOMETRICREPRESENTATIONSUBCONTEXT
97 IFCGEOMETRICREPRESENTATIONCONTEXT
94 IFCAXIS2PLACEMENT3D
6 IFCARTESIANPOINT
95 IFCDIRECTION
383329 IFCPOLYLINE
9 IFCARTESIANPOINT
383327 IFCARTESIANPOINT
383325 IFCSHAPE REPRESENTATION
102
IFCGEOMETRICREPRESENTATIONSUBCONTEXT
97 IFCGEOMETRICREPRESENTATIONCONTEXT
94 IFCAXIS2PLACEMENT3D
6 IFCARTESIANPOINT
95 IFCDIRECTION
383315 IFCEXTRUDEDAREASOLID
383311 IFCISHAPEPROFILEDEF
383310 IFCAXIS2PLACEMENT2D

```

Fig. 5. Sample graph subset of an IFCBEAM, I-Beam definition showing record ID and category. Taken from IFC exclusively representing structural entities.

process with remaining line segments, finally sorting the closed polys by area to separate the outer boundary and void representations.

Through linking Unity triangles with the Face.Triangle objects, the library only requires triangle calculation, if meshes are procedurally generated independent from any existing mesh. FaceSet to Unity entity conversion was achieved by iterating over the set and generating MeshFilters from replicated triangles. Facing was primarily created for basic parametric modelling. Although often frustrating, Unity Vector3 coordinates are read-only. This meant that any operation carried out on the Face could not be simplified where it is possible to carry it out directly on the existing mesh, the Unity mesh vertex list is recreated whilst the Face.PolyBoundary vertices were modified directly.

Void inner surfaces are used to identify nested windows and doors by finding their centre points and evaluating point-inside-boundary for XY and YZ compounded profiles. Linking between PolyBoundary and MeshFilter objects coupled with knowledge of envelopes and their embodied fixtures allows basic parametric modelling offsets to be achieved with relatively low time complexity. However, the library does not yet support void insertion to existing MeshFilter objects.

4.6. IFC model construction and Unity linking

Linking Unity and IFC entities was decided to be a run-time (rather than prep-time) or as a one-off event, such that bidirectional

transactions between models could be accommodated and functionality intended for post-completion extensions to the project would not require refactoring. Extending the IFC file and revision histories were considered as alternatives but were deemed unnecessary and over-complicated for the planned functionality.

Unity models differ from representations in IFC files in several ways:

- The naming convention changes depending on the way in which the model was generated and whether the Unity model was imported from .obj or .fbx.;
- Unity entities are discrete representations of IFC entities which lose precision and functional constraints (predominantly with arcs and other abstract geometric shapes);
- Unity's coordinate system is XZY where IFC is standard XYZ;
- data structures are changed from 128bit real as implemented with many IFC record properties as per ISO 16739:41 to 32bit float.

These prevent standard one-to-one mapping through name, geometry or spatial similarities and can require multiple validations to link entity definitions in both environments. Although indexing the IFC model was carefully constructed to reduce latency between identifying Unity entities and linking with their associated IFC model representations, it was not possible to maintain suitable framerate within the Unity environment synchronously which lead to semi-lazy, asynchronous linking. Revit families presented several additional complications including ambiguous naming which required further consideration of the information available in the IFC model.

4.6.1. BIM models without families

Naming was the primary linking mechanism for entities, where Unity import models were produced without families in the BIM interface export. Through testing imports with differing sources it was discovered that naming changes during Unity import were not consistent across all sources, notably differing where the import model was imported as .obj or .fbx. There were five recurring changes which were not a symptom of character encoding:

- **White space to colon:** White space characters whether space or tab were replaced with colons including multicharacter instances;
- **White space to underscore:** White space characters whether space or tab were replaced with underscores including multicharacter instance;
- **Case sensitivity:** Casing was not guaranteed to be retained between IFC and Unity models, although this may be a symptom of using intermediary software such as Twinmotion or 3DMax;
- **Multi-character replacement of colons and white space characters:** Several instances of substrings were replaced with either individual characters as per previously mentioned amendments, in some cases with characters in others, substrings;
- **Arbitrary inclusion or exclusion of square brackets:** Square brackets which were present in either IFC or Unity model where not guaranteed to be present in their complementary model entity names.

Due to these differences, a simple or partial string-matching process was not practical. The end solution for name-based linking procedurally generated regular expressions, which guaranteed either the name that was as expected or was at least consistent with one or more of the previously mentioned amendments.

Unity does not permit interactions with game objects through secondary threads preventing run-time referencing of game object from the linking thread. This constraint was remedied through extension of the CADObject to decouple useful properties from their Unity entities. Concurrency was retained through further extension to CADObject to incorporate entity modification functionality through its instances as an intermediary between Unity and IFC. This did however, require extension to the synchronous component of initialisation to generate

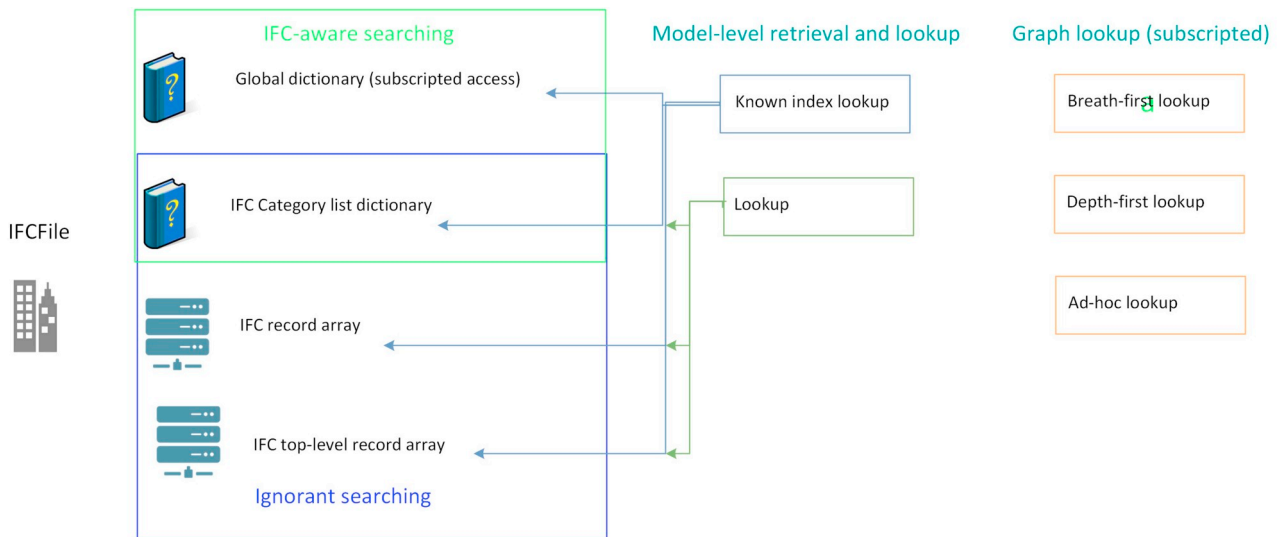


Fig. 6. IFCFile lookup indexing model.

CADObject instances and extract meaningful properties prior to threading.

4.6.2. BIM models with families

Building models exported with families proved more challenging due to game object nesting in the Unity environment as families are nested n-depth depending on the entity and context. For example, furniture may be nested import → model → family → type → object → entity compared to windows import → model → windows → entity. This meant that the project either needed to remove ignorance from the linking process or create a mechanism for identifying entities, where nesting depth and end entity naming was not guaranteed to be consistent with the IFC.

Removing ignorance would have resulted in failing to meet the secondary objective of this project so we created an alternative to the existing name linking method which constructed the family name by traversing the Unity model hierarchy upwards until the model entity was identified constructing a regular expression for family-equivalent lookups.

Construction of family lookup patterns was further complicated by redundant top-level name components which were relevant only within the context of the family set, not the IFC entity name. The final lookup method for models with this type of structure extended the family-less method to add lazy matching of top and bottom-level Unity game object names, which increased the worst-case lookup duration but was mitigated with top-level searching in place.

Familied models can result in several IFC candidates for individual Unity entities due to ambiguous naming in both models, which required additional filtering to retrieve the correct representation from the IFC model. After working through the IFC model, this research was unable to identify a match criterion that would accommodate linking without referencing geometric definitions leading to the first utilisation of the IFC-CARTESIANPOINT entity type. Given the IFC and Unity worlds are not identical, nor are their respective data structures. This meant this method would inevitably become intensive as the model complexity increased or instances of specific entities increased. The process would deviate from localising insertion points dealing with further discrepancies in the Unity environment. As a response to that, the study constructed a localisation method utilising bounding boxes such that centre points could be used without greedy translation of IFC geometry to Unity.

4.6.3. Strategy for linking Unity and IFC

Maintaining usable performance required numerous changes to the library, as functionality for linking and other operations were

introduced. These were resolved through refinement of IFC record indexing and dictionaries, increasing load on the parsing stage of the C# IFC model construction. These changes were later mitigated through replacing the regular expressions-based parser with a character-by-character explicit comparison mitigating the increased load from indexing, reducing parsing time significantly. The linking process is heavily dependent on a semi-eager approach to lazy evaluation which proved a common requirement throughout the IFC library development and working with Unity. However, the prototype presented here was able to achieve a link between both models without vendor-lock.

A flexible representation of the IFC format was created in C# to enable querying the IFC model based on the extent in which the IFC Category objects are defined in the referencing application. The model was initially designed with query performance at the centre of its focus which lead to several caching strategies being implemented. A dictionary was created for record ID queries, where the IFC record is known in advance, which became the subscriptable reference for the IFCFile class, a standard list was created for queries with either unknown related IFC records or those where multiple results are expected, and a dictionary of lists for each IFC category present in the model (Fig. 6).

The model was designed to be easily extended without modifying the core model construction for class-dependent constructors. Using C# meant that run-time object type delegation without hardcoding cases was not straightforward. The recommendation from the C# documentation is the use of the Activator class. However, the Activator's performance is impractically poorer than hardcoded delegation and could not be implemented for IFC files given their record counts are in millions. This study built an alternative using Linq and the ConstructorInfo class to cache the standard constructor for IFCObject descendant classes into a category-keyed dictionary for the CreateIFCObject factory. This approach did not suffer notable performance difference from the switch case alternative and remained a moderately low contributor to file parsing time.

4.6.4. Entity manipulation and IFC exporting

The IFCFile library supports manipulation of properties and classification information found in the base IFC model and is extensible to accommodate for manipulating or otherwise interacting with any information present in the model. The IFCFile model representation being graph-based scopes queries and property discoveries such that the developer can request a property with a known IFC* identifier and the graph traversal will ensure the correct instance is identified. The process has several variations which need to be known to the developer

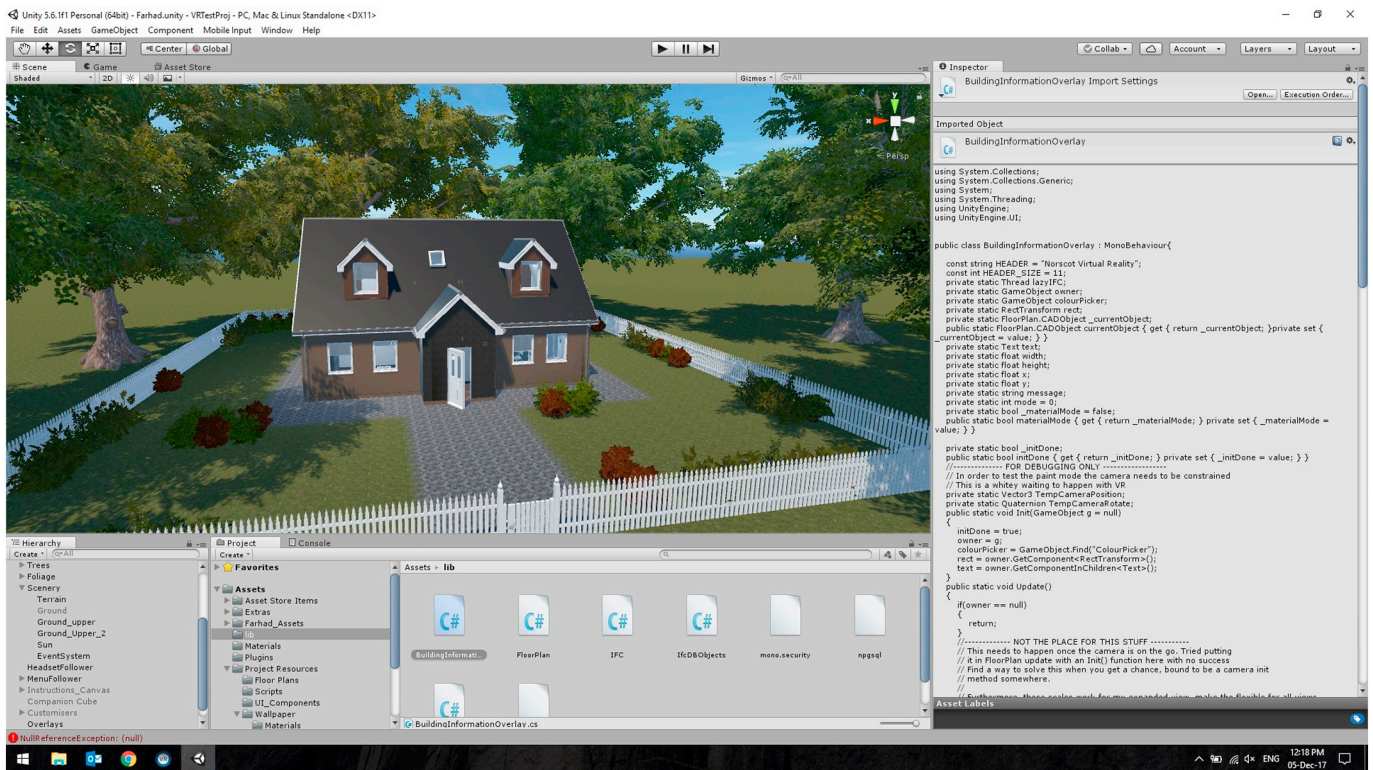


Fig. 7. Prototype development in Unity.

which determine the direction of the initial traversal, the desired target and element property index or indices required. Below summaries how to move and retrieving and updating the U-Value of a wall representation.

Moving the element (relative position):

- With the IFCWall:IFCObject either use the IFCFile[< # >] subscripted access method with the fifth property of the IFCWall object or retrieve the first IFCLOCALPLACEMENT via a depth search.
- With the IFCLOCALPLACEMENT either use the IFCFile[< # >] subscripted access method with the second property of the IFCLOCALPLACEMENT to retrieve the first result from either an ad-hoc or breadth search for IFCAXIS2PLACEMENT3D
- Apply arithmetic operation to the three sub properties of the only property in the IFCARTESIANPOINT identified either via IFCFile[< # >] subscripted access with the IFCAXIS2LOCALPLACEMENT3D's first property or retrieval of the only IFCARTESIANPOINT via breadth, depth or ad-hoc search.

Retrieving and updating wall element U-Value

- From the IFCWall:IFCObject call traverse up the tree to find the element's parents.
- Search the parent objects for the associated IFCRELDEFINESBYPROPERTY object.
- With the IFCRELDEFINESBYPROPERTY either use the IFCFile[< # >] subscripted access with the sixth property or retrieve the only IFCPROPERTYSET via a breadth, depth or ad-hoc search.
- With the IFCPROPERTYSET perform a breadth or ad-hoc search for the only IFCPROPERTYSINGLEVALUE objects and iterate over them comparing their first property to the string "ThermalTransmittance".
- Where an IFCPROPERTYSINGLEVALUE is found the U-Value is available and can be retrieved as the first sub property of the fourth property. To update this property simply replace the fourth property with the new U-Value

4.6.5. Model exporting

The BIMServer section best describes the bidirectional aspect of the IFCFile library whereupon actions are applied directly to the base IFC model as record updates or insertions. This means that where a change in the Unity environment which is translated to IFC is made, the corresponding IFC records are updated. 4.6.4 for example, outlines the process in which the U-Value of an entity may be updated where if the value of the example record 687 was updated from its starting value to 0.55 the change on the IFCObject and subsequent string representation in the IFC file would be as follows:

Record update:

lfc[687].properties [2].value = "IFCTHERMALTRANSMITTANCEMEASURE(0.55)"

Original (lfc[687].toString()):

687 IFCPROPERTYSINGLEVALUE('ThermalTransmittance',\$,IFCTHERMALTRANSMITTANCEMEASURE(1.),\$);

Amended (lfc[687].toString()):

687 IFCPROPERTYSINGLEVALUE('ThermalTransmittance',\$,IFCTHERMALTRANSMITTANCEMEASURE(0.55),\$);

5. Virtual showroom Interface

5.1. Integration strategy for VR

For the purposes of this project, Unity served as the main medium of application development as it allows for the easy running and compiling of applications from the user's personal computer. Unity was also chosen as it is free and easy to use, whilst offering quite a large online user base thus contributing to knowledge dissemination. Additionally, the shift from virtual to augmented reality could be achieved through a simple adjustment of the settings.

5.1.1. Controls in virtual reality

The primary VR headset that was used for this project was the HTC Vive. The HTC Vive comes equipped with two controllers and two wall

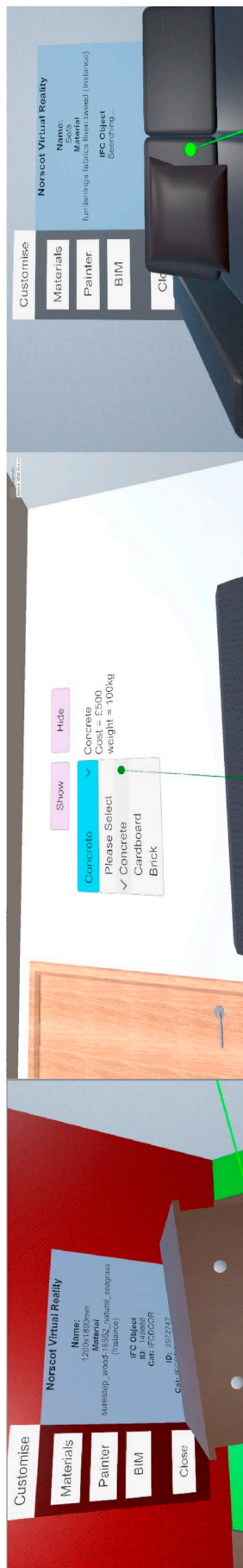


Fig. 8. Interaction with BIM data.

mountable scanners that allow the user to move around extensively within their personal real-world space. The Vive's controllers consist of five buttons, four of which ('Menu', 'Touchpad', 'Grip', and 'Trigger') supported the assignment of custom functions. There were however, more functions that required buttons than there were buttons available, meaning some functions had to be restricted to one specific controller – left or right – for situational purposes.

The 'Menu' button on either controller was used to access the on-screen teleport menu, which was displayed in the form of a floor plan of the building. On both controllers, the 'Touchpad' was used to activate a pointer to interact with the virtual environment. On the left-hand controller, the touchpad was used to 'teleport' or move around the virtual world, whereas the right-hand controller's touchpad was used to interact with user interface elements. The 'Grip' button was used to move chosen furnishings.

5.1.2. Movement in virtual reality

This study explored several movement control mechanisms. The method applied varied depending on how the virtual showroom works and expected interactions with the environment. Movement driven by real-world movement is very intuitive to enact albeit the user is tethered to a computer with the VR headset and wires. Freedom of movement in the real-world space can also vary due to furniture or walls but is always limited with current technologies. Within the Norscot Virtual Showroom, the user can traverse an entire house, as well as a small section of the exterior garden with a fence acting as an environment boundary. Fig. 7 illustrates the overall view of the prototype, as well as developers' view in Unity.

A "teleport" function was added such that users could traverse the showroom without physical movement. This accommodated movement which may otherwise have been constrained by the real-world. This solution also aimed to reduce motion sickness. If, for example a button was added that allowed the user to 'walk' in VR without moving in the real world, this would cause motion sickness as the user sees themselves moving in VR but would not physically feel like moving in the real world.

5.1.3. Interfaces in virtual reality

Diegetic Interfaces within VR applications can be complicated to get right and can often cause many issues. Diegetic interfaces are interfaces that exist within the world or environment and are not entirely obvious or 'in your face' such as a light switch button on a wall. Non-diegetic interfaces such as Heads Up Display (HUD) which covers the entire screen, can be found in many games and apps. The most important reason for keeping interfaces diegetic in VR is so that the users' vision is not 'blocked' by an interface, as there is only a limited field of view available when using a VR headset. Alongside this, having diegetic interfaces means that the user is more immersed in the surrounding virtual world without being disrupted by a wall of text, icons, and images.

Many VR apps also keep interfaces confined to the controllers, where users can simply lift their hand, look at the controller and see one primary interface that allows them to carry out all the actions they need. This is more effective than keeping interfaces on walls or on the ground, as now the user has the choice of seeing an interface or not, much like a person using a watch on their wrist.

The difference between the objects that are interactive and those that are not is difficult to present in a VR setting. Designers of VR apps can employ many strategies that range from an obvious user interface hovering over whatever object is interactive to highlighting an object only when users 'touch' it with their controller. To maintain immersion within this prototype, no interfaces were used to indicate if an object was moveable, meaning objects such as furniture that were moveable have their outline highlighted with a certain colour when the user points at the object with their controller's pointer. This allows the user to easily differentiate between what is interactive and is not without the

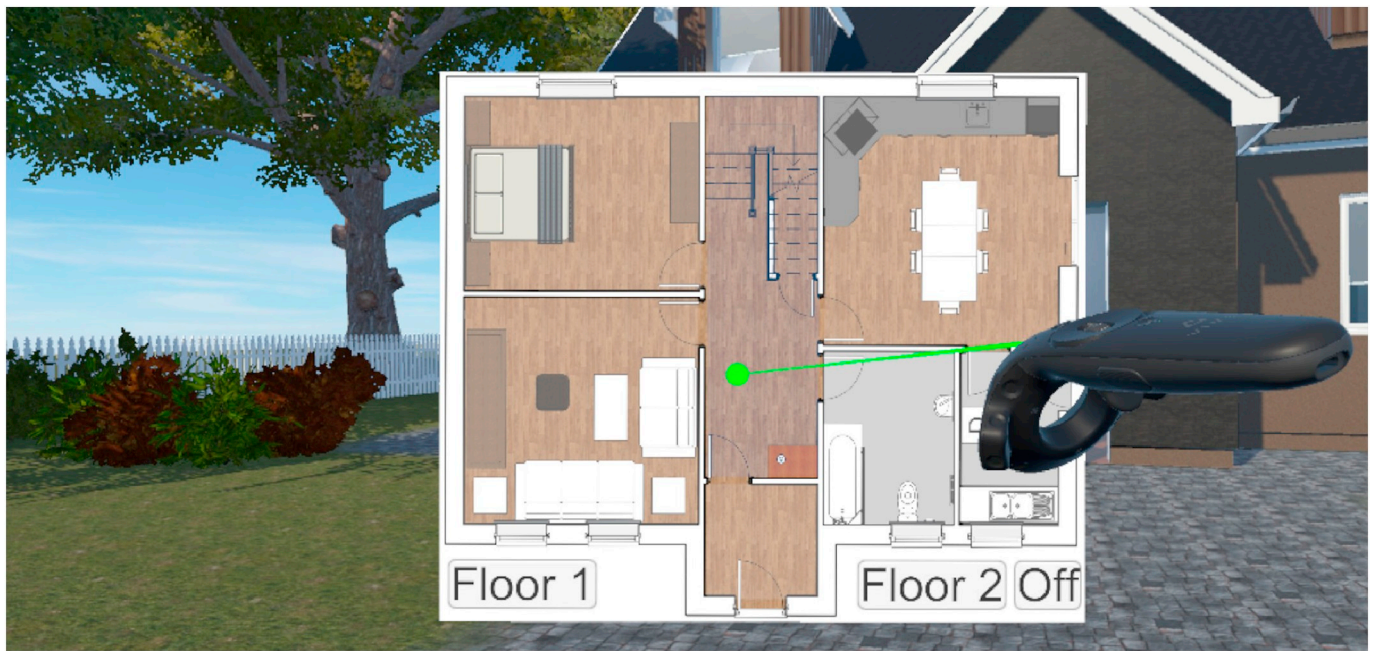


Fig. 9. Teleportation Interface in virtual reality.

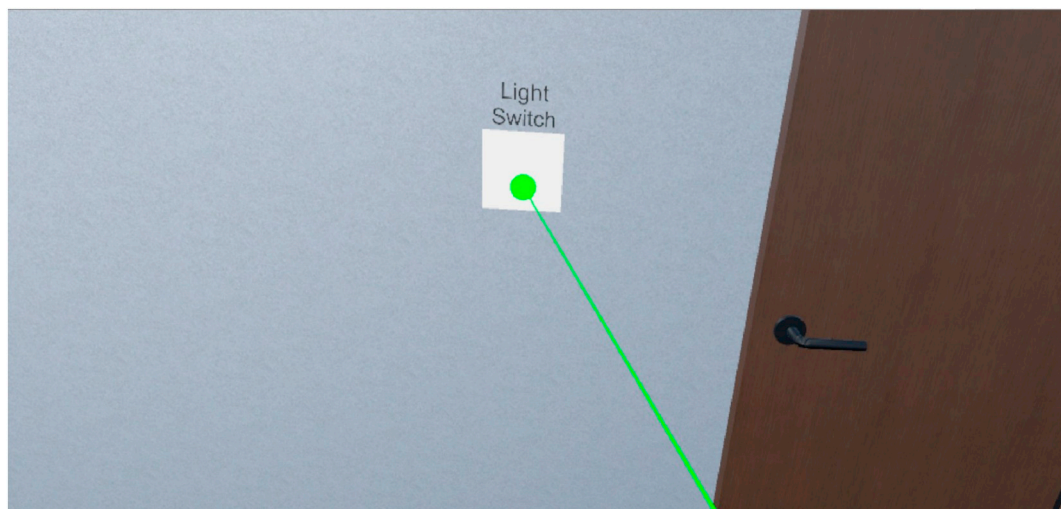


Fig. 10. Light switch interface.

need to read the entire interface. Likewise, interfaces such as the material picker, colour painter, and light switches have buttons that are clearly visible to the user.

Another problem that faces many designers of VR applications is that text can often be quite hard to read in VR. The user must move their head around to read text, rather than just their eyes, and the user's distance from the piece of text can also affect the visual quality. In this project, the text was only contained within interfaces such as the material picker, and IFC interface, which mostly consisted of trigger buttons or drop-down menus (Fig. 8).

Teleport Menu: The user could press the “Menu” button which would then activate a floor plan teleport menu (Fig. 9). This interface would display a floor plan of the house and from here the user could select a level and click a specific room to teleport into. This interface was attached to the head of the user within the application. By placing the interface on the head of the user, they would easily identify and use the interface without losing it. One issue that arose with this interface was that because the interface was placed a set distance away from the

head of the user, if they were very close to a wall, the interface would often get lost behind it. As the building used for the prototype was quite confined, this issue occurred often during usability testing, nonetheless a solution was found to stop the interface from rendering behind any objects.

Light Switch: This was a simple interface consisting of an easily recognisable white box with text above labelled “Light Switch” (Fig. 10). The user could simply point and click to turn lights on and off. This interface was placed on walls in areas where one may typically find a light switch in order to allow for immersion and not create an intrusive object. In other words, the interface only ‘exists’ when the user is looking for it or just happens to spot it.

Customisation Menu: This was the most extensive interface, accessible by clicking the trigger button on the left controller. This interface was placed on every wall which could be customised so that the user could choose specifically how they wanted to decorate each wall. The interface, by default, was hidden to allow the users to immerse themselves in the showroom first, and then activate and deactivate the

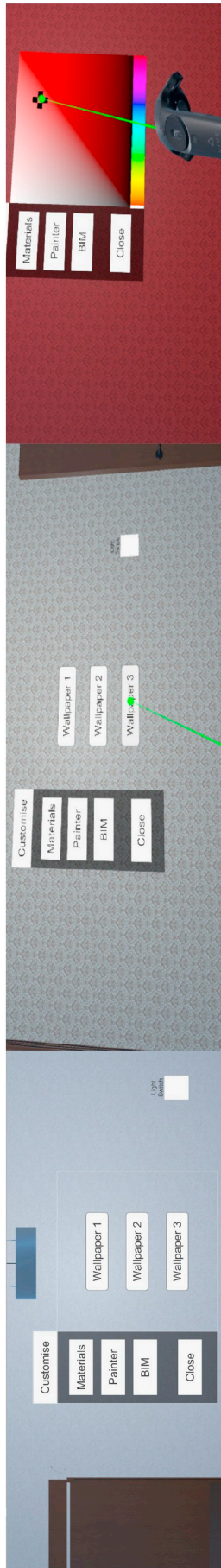


Fig. 11. Interaction with wall finishes in the immersive environment.

interface as they wished. This meant that the users had no interface permanently plastered on to a wall that would not allow them to fully view it. The freedom of transition between interface and no interface also helped to allow the user to exist in two different game states - immersion and customisation. Fig. 11 shows some examples of interaction with wall finishes in an immersive environment.

5.2. Integration strategy for AR

The Google Tango platform was used to run the augmented reality experience of the offsite manufactured kit home showroom prototype. Currently, the application is only supported by certain mobile devices, so the *Lenovo Phab 2 Pro* was used for this project. Tango is a mobile application developed by Google that makes use of the two cameras on the back of the mobile device in order to take accurate measures of 'motion tracking' and 'depth perception' which then provides a smooth and engaging augmented reality experience. Tango also offers an 'area learning' feature that remembers where objects are placed in relation to the users' position which prevents them from being lost or 'drifting' away from the users when they turn and walk in a different direction or hold the device at an awkward angle.

As part of the transition from BIM to immersive experiences, the 3D model of the house, already imported in Unity for virtual reality, could simply be re-imported into a separate augmented reality enabled version of Unity. This split of the flow of immersive application development is necessitated by the fact that Unity does not support virtual and augmented reality simultaneously, whilst also being compiled for two separate platforms - desktop and mobile. Nonetheless, the information present in both VR and AR is consistent.

5.2.1. Controls and interfaces in augmented reality

Due to the nature of mobile applications and touchscreens, and lack of controllers as opposed to VR, controls and interfaces within augmented reality need to be kept mostly as non-diegetic. This meant that there were multiple buttons on the screen which would carry out similar actions to that of the VR version. Because of this, some immersion was lost. However, this was not detrimental as the interface was created to be aesthetically subtle. Some form of interface was required regardless as the mobile phone's touchscreen was the only appropriate form of input to perform actions within the showroom environment. The controls consisted of a teleport menu, zoom in/out, and move forward/backward buttons. The AR version of the virtual showroom also featured the same customiser and light switch interface as the VR version bound to internal partitions.

5.2.2. Movements in augmented reality

The users had more freedom of movement than in the VR version as it does not require tracking scanners. This did not mean however that walls and furniture in the physical world were not a problem, as real-world obstacles could still block the users from reaching their desired location within the showroom. For this reason, a teleport menu was also added, consisting of one "Teleport Menu" button could expand into several buttons, each signifying a teleport to the centre point of every different room in the showroom (Fig. 12). This feature was also useful in situations where users got lost or confused, by helping them 'reset' their location. In addition to this, the "Move Forward" and "Move Backward" buttons moved the user to a set distance forward or backward. This could be useful whenever a user was blocked from reaching their desired location due to the real-world obstacles. The "Zoom in" and "Zoom Out" button was implemented to allow users to adjust their camera's field of view. The feature was added based on the users' feedback during usability test trials (Section 6).

5.2.3. Interactions in the AR application

As there was no 'pointer' in the AR interface, portability of objects was not as readable for users as it was in the VR. In other words, users



Fig. 12. Mobile version of the showroom and teleportation menu in AR.



Fig. 13. Specific setting and environment of the conducted usability tests.

only had to ‘guess’ which objects were moveable and tap and hold on them. This was not so much of an issue as non-static objects were made moveable in this version of the showroom. Once the user tapped on a moveable object, it would then hover in the air allowing them to position it to a desired location by physically walking, whilst still holding the object. This study found through the feedback from users that trying to find out which objects were moveable allowed them to experience a feeling of experimentation and higher sense of immersion.

6. Usability test

At the final stage, the prototype was tested, validated and refined by conducting a series of usability test trials. These trials engaged with the project 20 participants from different age groups and varied VR/gaming experience levels. 9 participants were selected from designers, managers and customers of Norscot, who volunteered themselves to contribute to the study. 11 more participants were invited to attend the test during the Scottish Innovation of The Year 2018 Award Ceremony. All individual and group briefing tests started with a general discussions regarding the project, followed by a 10 min detailed briefing for every participant to familiarise them with the technology and train them to use the hardware correctly. Each participant was then given 10 min to experience the virtual showroom navigating through different spaces and experiencing different features of the software, whilst one team member was chatting with them throughout and collecting verbal feedback. Once the tests were complete, each participant was invited to complete a usability test questionnaire comprising of 10 questions regarding their experience from using the system. For each question, the users were asked to rank their experience from using the specific feature between 1 (Difficult-Confusing) and 5 (Very Easy-Intuitive). The range of questions varied from the ease of teleporting through the

showroom, or climbing the staircase, to customising the interior such as changing the type of wallpaper or manipulating features imported from the IFC file, such as U-Values or types of windows. Finally, each user was requested to answer an open ended question regarding their overall experience and any specific challenges that they thought should be addressed in the next stages. Fig. 13 shows the specific setting and environment of the conducted usability tests.

Regarding the VR environment, all users appreciated the developed VRE and considered it an engaging way to show how a house would look and feel. Movements and interaction inside the showroom were found quite smooth, and no user raised issues of ‘motion sickness’ or ‘jittering’. Participants did not also struggle with adapting back to normal environments after the testing was complete. With regards to the interactions implemented in the VR environment, all participants considered turning lights on/off easy and intuitive. Wallpaper changes were also straightforward, whilst paint colour change was less obvious.

The study also found through the tests that some parts of tutorial sections for VR environment (Fig. 14) were too heavy with text and these were proven hard to read by many participants. It was also found that many users want to rush into the action rather than wait and complete sections of tutorials. For this reason, these were replaced with graphics.

Further details of the provided feedback by the participants and the solutions suggested by the study are provided in Table 1.

Regarding the AR environment, feedback from the users as well as clients indicated that they would like for the app to be usable on iOS (a firmware used in iPhone devices) and mainstream Android platform. In other words, they were not too much particular on the extra capabilities that Tango could offer to them, such as possibility of physically walking in order to navigate in the virtual showroom, whilst they mentioned these as very useful functions. The only remark on the functionality of

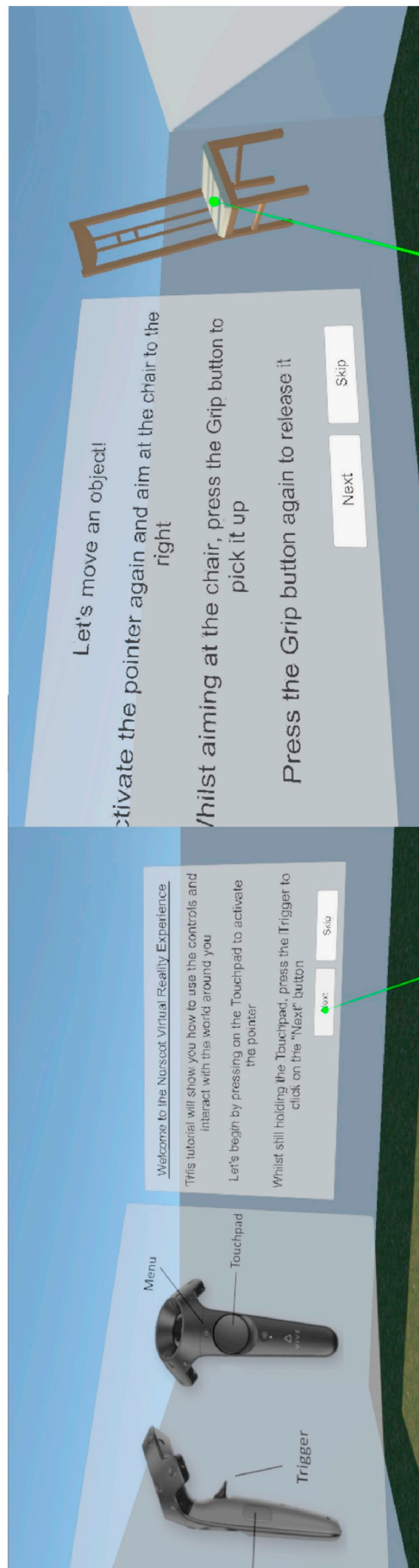


Fig. 14. Tutorial section of the VR application.

the AR showroom concerns the 2-button Zoom function to change the camera's field of view, which appeared hard to use. This study provided a potential solution to this issue by allowing users to just use finger pinch movement to zoom in and out or display the value of the field of view number to the user as an additive piece of information. Overall, besides giving confidence to the researchers regarding the quality of the developed prototype, some useful feedback was collected from the participants and applied to improve the showroom interface, as per discussed in this section as well as previous sections of this paper.

7. Conclusion

This paper demonstrated the feasibility of creating an abstract standardised interface for IFC modelling which can be utilised by an interfacing application. The research presented in this paper addresses the functionality gap between BIM and immersive technologies through an interface that integrates 3D building information into an immersive environment. As became evident in the reviewed literature, despite the wide adoption of BIM in the AEC industry, there are still data management and software interoperability issues that impede the smooth collaboration between AEC professionals. Interoperability is long known to be a significant factor in spurring innovation and competition among vendors. The interoperability and bidirectional achievement demonstrate that BIM integration need not be constrained to a particular proprietary interface nor require the interfacing application be it a strict vector CAD package or even a discrete geometry virtual environment.

This study is a response to the need for a universal platform that streamlines design processes and engages clients, especially those untrained in spatial comprehension, as is often the case in the house building sector. This project extended the synergy between technologies with the integration of immersive technologies that tackle the challenges of perception imposed by conventional drawings. VR representation of BIM models also improved the immersion into building designs and facilitated the comprehension of complex spatial problems related to physical form and non-physical constraints.

Despite many good attempts by other studies for developing VR add-ons for the major BIM applications, especially Autodesk Revit, there is little previous research to integrate openBIM with VR environments. This study specifically aimed at this integration, since leveraging interoperability was its overarching aim. Another innovative element in this project lay in the shift away from bulky and expensive headset-based VR systems towards cheap and user-friendly Google Tango interface, which has capability of remembering the space and tracking users' move in 3D, as well as integrating the physical environment with VR. The project is also unique in terms of linking VR representation to openBIM information in real-time, offering data-rich virtual showroom to the users, which provides support from early decision making to the latest O&M stages. The fact also that this showroom is available both on VR and AR platforms can enable more distant users to engage with and help them use it in a way that suits their needs.

Developing such powerful interactive design and visualisation applications can help Small and Medium Enterprises (SMEs) address growing customer/market expectations. By giving the client (homeowner, self-builder or house builder) an interactive and immersive design environment, which enables them to see inside the home and truly visualise the environment, they will be better able to make design decisions and are more likely to proceed to purchase. Thus, such virtual showrooms can be used for both marketing and streamlining of the purchasing process, ensuring homes are delivered to clients' expectations on a right first-time basis. This can also streamline sales processes by minimising costs for clients and the businesses (minimal design changes/iterations prior to sign off). The interface of such showroom does not necessarily replicate the functionality of existing BIM platforms, but instead offers clients the ability to combine advanced

Table 1
Virtual reality mode usability test feedback.

Feedback	Solution(s)
1) Users could get confused between the left and right controllers	• Change the models of the controllers in VR so that users can visually differentiate between them
2) Client wanted the colour picker/gradient to be changed to a 'swatch' interface	• Additionally, interfaces can be taken from the world and added to one of the controllers
3) Interface elements such as buttons & text were too 'basic'	• Add an interface of buttons that display the colour (and possibly names) of set paint colours. These colours/materials could be loaded from a library
4) Tutorial text was quite hard to read and understand	• A custom interface and supporting elements (buttons/fonts) could be created in a 3rd party software such as Adobe Photoshop/Illustrator
5) Grip and Touchpad buttons were hard to find due to controllers' design	• Have an assistant to help user understand the controls and app's functionalities
6) Using touchpad alone is easy, but added functionality creates confusion in movement/interfaces	• Have an assistant to help users understand the controls before using the app
7) Menu and Teleport button were often mixed up	• Add some form of animation or highlighting on a button in VR to visually indicate the buttons
8) Teleport Menu (Floor Plan Menu) was hard to see in tight spaces – got lost inside the walls	• Have an assistant to help users understand the controls before using the app
9) Colour Picker/Painter interface was hard to use with head movement	• Add some form of animation or highlighting on a button in VR to visually indicate the buttons
10) Holding down the Touchpad whilst using the Trigger was the most common issue	• Add this interface to the controller rather than the headset
11) Head set is quite heavy, and cables can be a nuisance	• The gradient interface will be changed to swatches so head movement for a gradient will no longer be required
12) Teleporting with the controller led to some participants teleporting into walls	• The controller's pointer can be made to automatically activate when hovering over an interface, rather than manual input
13) Some participants were concerned about the headset messing up their hair/make up	• Certain current and future headsets feature wireless virtual reality capabilities
14) Some participants noted that it was quite warm wearing the headset, a possible issue in summer months	• Allow the pointer to only recognise the floor as a teleport location
15) Gravity on moveable objects did not work	• Other objects require colliders, so 'tags' can be applied within Unity to create exceptions
16) Some participants struggled to identify the room types on the floor plan	• N/A
17) Unable to adjust the distance of a moveable object	• N/A
	• Caused by all objects becoming "rigid bodies"
	• Exceptions will need to be created for moveable objects to allow for gravity
	• Add names on the floor plan image to allow users to identify each room
	• Add a button/function on the controller that allows the user to change the distance of an object

interactive visualisation with the power of BIM. Architects and house builders, in addition to benefiting from the above marketing and workflow tool, have access to all relevant BIM data and can see the impact of design changes on building performance.

A fully working proof-of-concept prototype was produced, which includes all aspects of integration between openBIM, VR and the Tango environments. The next stage will be taking the application to a commercial level and providing an automatic platform to import various kinds of BIM files and set up a common user environment directly from IFC file. Yet, commercialisation of this project will be highly dependent on further development of BIM product libraries, which are still in their infancy and need prolonged engagement with industry and users to explore their real impacts and implications.

This project is in a right trajectory for the aforementioned commercialisation aim, since the developed proof-of-concept-prototype is an agnostic pared-down openBIM system that does not discriminate between BIM authoring tools. Here, the building information taken straight from the authoring tool is transformed through an IFC conversion into a database that is then visualised on Unity platform, as a game engine. One of the main contributions of this project was providing practical exemplar for agnostic BIM interfacing in order to enable development of client applications utilising any COM supporting interface, such that media richness is constrained by the creativity of developers rather than existing proprietary BIM packages, e.g. Autodesk Revit. The BIM server application developed in this study on the other hand, proves the concept and demonstrates the basic functionality necessary for concurrent multiuser, low-latency openBIM interactivity and data concurrency. This is a significant step towards reducing psychological distality, which is known to be a major contributor to lack of effective inter-organisational collaboration along with low-latency data sharing. In its current state, the developed prototype provides practical development guidelines however, it must be admitted that its abstract nature led to programmer-focused functionality. Future development should focus on social presence and effective balancing of concurrent entity interaction and telepresence.

Acknowledgement

The authors would like to gratefully acknowledge the generous funding received from Innovate UK, Scottish Funding Council, Construction Scotland Innovation Centre (CSIC), and Norscot Joinery Ltd for this award-winning project which eventually received the buildingSMART International Awards 2018: SPECIAL DISTINCTION and Scottish Knowledge Exchange Awards 2018: THE INNOVATION OF THE YEAR AWARD.

References

- [1] F.H. Abanda, J.H.M. Tah, F.K.T. Cheung, BIM in off-site manufacturing for buildings, *J. Build. Eng.* 14 (Supplement C) (2017) 89–102, <https://doi.org/10.1016/j.jobe.2017.10.002>.
- [2] K. Afshari, C.M. Eastman, D. Castro-Lacouture, JavaScript Object Notation (JSON) data serialization for IFC schema in web-based BIM data exchange, *Autom. Constr.* 77 (Supplement C) (2017) 24–51, <https://doi.org/10.1016/j.autcon.2017.01.011>.
- [3] Y. Arayici, T. Fernando, V. Munoz, M. Bassanino, Interoperability specification development for integrated BIM use in performance based design, *Autom. Constr.* 85 (Supplement C) (2018) 167–181, <https://doi.org/10.1016/j.autcon.2017.10.018>.
- [4] S.A. Assaf, S. Al-Hejji, Causes of delay in large construction projects, *Int. J. Proj. Manag.* 24 (4) (2006) 349–357, <https://doi.org/10.1016/j.ijproman.2005.11.010>.
- [5] S. Banihashemi, A. Tabadkani, M.R. Hosseini, Integration of parametric design into modular coordination: a construction waste reduction workflow, *Autom. Constr.* 88 (2018) 1–12, <https://doi.org/10.1016/j.autcon.2017.12.026>.
- [6] K. Blind, *The impact of standardization and standards on innovation: compendium of evidence on the effectiveness of innovation policy intervention*, Nesta Working Paper 13/15, Manchester Institute of Innovation Research, Manchester, UK, 2013.
- [7] D. Cao, G. Wang, H. Li, M. Skitmore, T. Huang, W. Zhang, Practices and effectiveness of building information modelling in construction projects in China, *Autom. Constr.* 49 (Part A) (2015) 113–122, <https://doi.org/10.1016/j.autcon.2014.10.014>.
- [8] H.S. Cha, D.G. Lee, A case study of time/cost analysis for aged-housing renovation using a pre-made BIM database structure, *KSCE J. Civ. Eng.* 19 (4) (2015) 841–852, <https://doi.org/10.1007/s12205-013-0617-1>.
- [9] H.-L. Chi, S.-C. Kang, X. Wang, Research trends and opportunities of augmented reality applications in architecture, engineering, and construction, *Autom. Constr.* 33 (Supplement C) (2013) 116–122, <https://doi.org/10.1016/j.autcon.2012.12.017>.
- [10] J. Choi, I. Kim, Development of openBIM-based interoperability system for code

- checking system, Green and Smart Technology, 2016 2016.
- [11] J. Choi, I. Kim, A methodology of building code checking system for building permission based on openBIM, ISARC 2017 - Proceedings of the 34th International Symposium on Automation and Robotics in Construction, vol. 34, Vilnius Gediminas Technical University, Department of Construction Economics & Property, 2017.
 - [12] F. Craveiro, H.M. Bartolo, A. Gale, J.P. Duarte, P.J. Bartolo, A design tool for resource-efficient fabrication of 3d-graded structural building components using additive manufacturing, *Autom. Constr.* 82 (Supplement C) (2017) 75–83, <https://doi.org/10.1016/j.autcon.2017.05.006>.
 - [13] T. Damen, M. MacDonald, T. Hartmann, R. Di Giulio, P. Bonsma, K. Luig, R. Sebastian, D. Soetanto, BIM based collaborative design technology for collective self-organised housing, 40th IAHS World Congress on Housing, Sustainable Housing Construction, 2014, pp. 16–19.
 - [14] T. Damen, R. Sebastian, M. MacDonald, D. Soetanto, T. Hartmann, R. Di Giulio, P. Bonsma, K. Luig, The application of BIM as collaborative design technology for collective self-organised housing, *Int. J. 3-D Inf. Model. (IJ3DIM)* 4 (1) (2015) 1–18, <https://doi.org/10.4018/IJ3DIM.2015010101>.
 - [15] S. Dong, C. Feng, V.R. Kamat, Sensitivity analysis of augmented reality-assisted building damage reconnaissance using virtual prototyping, *Autom. Constr.* 33 (Supplement C) (2013) 24–36, <https://doi.org/10.1016/j.autcon.2012.09.005>.
 - [16] J. Du, Z. Zou, Y. Shi, D. Zhao, Zero latency: real-time synchronization of BIM data in virtual reality for collaborative decision-making, *Autom. Constr.* 85 (2018) 51–64, <https://doi.org/10.1016/j.autcon.2017.10.009>.
 - [17] U. Gasser, J.G. Palfrey, Breaking Down Digital Barriers: When and How ICT Interoperability Drives Innovation, Berkman Center Research Publication Social Science Research Network (SSRN) 2007-8, 2007, <https://doi.org/10.2139/ssrn.1033226>.
 - [18] N. Haile, J. Altmann, Evaluating investments in portability and interoperability between software service platforms, *Futur. Gener. Comput. Syst.* 78 (2018) 224–241, <https://doi.org/10.1016/j.future.2017.04.040>.
 - [19] N. Jaffar, A.H.A. Tharim, M.N. Shuib, Factors of conflict in construction industry: a literature review, *Procedia Eng.* 20 (2011) 193–202, <https://doi.org/10.1016/j.proeng.2011.11.156>.
 - [20] A. Khalili, D.K.H. Chua, IFC-based framework to move beyond individual building elements toward configuring a higher level of prefabrication, *J. Comput. Civ. Eng.* 27 (3) (2013) 243–253, [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000203](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000203).
 - [21] C. Koch, M. Neges, M. König, M. Abramovici, Natural markers for augmented reality-based indoor navigation and facility maintenance, *Autom. Constr.* 48 (Supplement C) (2014) 18–30, <https://doi.org/10.1016/j.autcon.2014.08.009>.
 - [22] Y.-C. Lee, C.M. Eastman, W. Solihin, Logic for ensuring the data exchange integrity of building information models, *Autom. Constr.* 85 (Supplement C) (2018) 249–262, <https://doi.org/10.1016/j.autcon.2017.08.010>.
 - [23] E.I.A. Lester, Chapter 42 - Conflict Management and Dispute Resolution, *Project Management, Planning and Control*, seventh edition, Butterworth-Heinemann, 2017, pp. 393–399, <https://doi.org/10.1016/B978-0-08-102020-3.00042-5>.
 - [24] X. Li, P. Wu, G.Q. Shen, X. Wang, Y. Teng, Mapping the knowledge domains of Building Information Modeling (BIM): a bibliometric approach, *Autom. Constr.* 84 (2017) 195–206, <https://doi.org/10.1016/j.autcon.2017.09.011>.
 - [25] H. Liu, M. Al-Hussein, M. Lu, BIM-based integrated approach for detailed construction scheduling under resource constraints, *Autom. Constr.* 53 (2015) 29–43, <https://doi.org/10.1016/j.autcon.2015.03.008>.
 - [26] H. Liu, G. Singh, M. Lu, A. Bouferguene, M. Al-Hussein, BIM-based automated design and planning for boarding of light-frame residential buildings, *Autom. Constr.* 89 (2018) 235–249, <https://doi.org/10.1016/j.autcon.2018.02.001>.
 - [27] S. Mitkus, T. Mitkus, Causes of conflicts in a construction industry: a communicational approach, *Procedia Soc. Behav. Sci.* 110 (2014) 777–786, <https://doi.org/10.1016/j.sbspro.2013.12.922>.
 - [28] G.P. Monizza, E. Rauch, D.T. Matt, Parametric and generative design techniques for mass-customization in building industry: a case study for glued-laminated timber, *Procedia CIRP* 60 (Supplement C) (2017) 392–397, <https://doi.org/10.1016/j.procir.2017.01.051>.
 - [29] N. Norouzi, M. Shabak, M.R.B. Embi, T.H. Khan, The architect, the client and effective communication in architectural design practice, *Procedia Soc. Behav. Sci.* 172 (2015) 635–642, <https://doi.org/10.1016/j.sbspro.2015.01.413>.
 - [30] A.M. Odeh, H.T. Battaineh, Causes of construction delay: traditional contracts, *Int. J. Proj. Manag.* 20 (1) (2002) 67–73, [https://doi.org/10.1016/S0263-7863\(00\)00037-5](https://doi.org/10.1016/S0263-7863(00)00037-5).
 - [31] R. Oxman, Thinking difference: theories and models of parametric design thinking, *Des. Stud.* 52 (Supplement C) (2017) 4–39, <https://doi.org/10.1016/j.destud.2017.06.001>.
 - [32] J. Park, BIM-based parametric design methodology for modernized Korean traditional buildings, *J. Asian Archit. Build. Eng.* 10 (2) (2011) 327–334, <https://doi.org/10.3130/jaabe.10.327>.
 - [33] P. Pauwels, S. Zhang, Y.-C. Lee, Semantic web technologies in AEC industry: a literature overview, *Autom. Constr.* 73 (Supplement C) (2017) 145–165, <https://doi.org/10.1016/j.autcon.2016.10.003>.
 - [34] E.A. Petrova, M. Rasmussen, R.L. Jensen, K. Svidt, Integrating Virtual Reality and BIM for End-user Involvement in Building Design: a case study, *The Joint Conference on Computing in Construction (JC3) 2017*, 2017, pp. 699–709, <https://doi.org/10.24928/JC3-2017/0266>.
 - [35] M. Rasmussen, A.N. Gade, R.L. Jensen, Bridging the Gap Between Actors and Digital Tools in a Knotworking Design Process, *When Social Science Meets BIM and LEAN*, vol. 1, (2017), p. 3 Chapter 5. Aalborg.
 - [36] J. Ren, Y. Liu, Z. Ruan, Architecture in an age of augmented reality: applications and practices for mobile intelligence BIM-based AR in the entire lifecycle, *DEStech Transactions on Computer Science and Engineering (iceiti)*, 2016, <https://doi.org/10.12783/dtcse/iceiti2016/6203>.
 - [37] M. Sarhadi, Comparing communication style within project teams of three project-oriented organizations in Iran, *Procedia Soc. Behav. Sci.* 226 (2016) 226–235, <https://doi.org/10.1016/j.sbspro.2016.06.183>.
 - [38] R. Sebastian, Integrated design and engineering using building information modelling: a pilot project of small-scale housing development in the Netherlands, *Archit. Eng. Des. Manag.* 6 (2) (2010) 103–110, <https://doi.org/10.3763/aedm.2010.0116>.
 - [39] Smart BIM Solutions, SmartBIM Solutions Open BIM & Closed BIM, (2012).
 - [40] D. Walasek, A. Barszcz, Analysis of the adoption rate of building information modeling [BIM] and its return on investment [ROI], *Procedia Eng.* 172 (Supplement C) (2017) 1227–1234, <https://doi.org/10.1016/j.proeng.2017.02.144>.
 - [41] X. Wang, M.J. Kim, P.E.D. Love, S.-C. Kang, Augmented reality in built environment: classification and implications for future research, *Autom. Constr.* 32 (2013) 1–13, <https://doi.org/10.1016/j.autcon.2012.11.021>.
 - [42] X. Wang, M. Truijens, L. Hou, Y. Wang, Y. Zhou, Integrating Augmented Reality with Building Information Modeling: onsite construction process controlling for liquefied natural gas industry, *Autom. Constr.* 40 (Supplement C) (2014) 96–105, <https://doi.org/10.1016/j.autcon.2013.12.003>.
 - [43] G. Wu, C. Liu, X. Zhao, J. Zuo, Investigating the relationship between communication-conflict interaction and project success among construction project teams, *Int. J. Proj. Manag.* 35 (8) (2017) 1466–1482, <https://doi.org/10.1016/j.ijproman.2017.08.006>.
 - [44] Z. Yuan, C. Sun, Y. Wang, Design for Manufacture and Assembly-oriented parametric design of prefabricated buildings, *Autom. Constr.* 88 (2018) 13–22, <https://doi.org/10.1016/j.autcon.2017.12.021>.
 - [45] B.G. Zulch, Communication: the Foundation of Project Management, *Procedia Technol.* 16 (2014) 1000–1009, <https://doi.org/10.1016/j.protcy.2014.10.054>.